



MANUAL TÉCNICO SL_001



Versión del sistema: 1.00
Fecha elaboración: 14/04/2017
Área de elaboración: Aula UAS.

INTRODUCCIÓN

En el siguiente manual se explican los aspectos más importantes en el diseño de del proyecto Sexy Lady, Dentro del presente manual, se encuentra una serie de diagramas que le darán a conocer la forma lógica del funcionamiento del sistema que ayudara a su mejora y aprovechamiento del mismo.

Se recomienda consultar el manual de uso de las siguientes tecnologías web como: PHP, SGBD MySQL, CodeIgneter, y el servidor web byte.host, para tener una mejor noción de uso acerca de la aplicación web “Sexy Lady”.

El presente proyecto está fundamentado en una serie de documentación generada durante el desarrollo del proyecto, y regida bajo CMMI y las normas ISO 9001, IEEE 830, IEEE 829, ISO 27000.

CONTENIDO

INTRODUCCIÓN

INTRODUCCIÓN	
OBJETIVOS GENERALES Y ESPECÍFICOS DEL SISTEMA.....	5
NORMAS, POLÍTICAS Y PROCEDIMIENTOS	7
IEEE 830.....	7
IEEE 829.....	7
ISO 27000.....	7
DEFINICIÓN DE LAS REGLAS DEL NEGOCIO IMPLEMENTADAS	8
FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA.....	9
DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	10
ESPECIFICACIÓN DE REQUISITOS	11
VISTA FUNCIONAL.....	15
VISTA LÓGICA.....	33
MODELO LÓGICO DE DATOS	46
DISEÑO DE PANTALLAS	47
DESCRIPCIÓN DE CAMPOS REQUERIDOS POR PANTALLA	51
VISTA DE IMPLEMENTACIÓN	55
VISTA DE DESPLIEGUE.....	56
DIAGRAMA DE NAVEGACIÓN DEL SISTEMA	57
GLOSARIO DE TÉRMINOS	58
ESTÁNDARES DE ELABORACIÓN DEL MANUAL	63

INTRODUCCIÓN

El sitio web permite brindar mayor publicidad a la empresa, además permite a la empresa incorporarse a las ventas online la cual es su principal función, el sitio web cuenta con el módulo de inicio, conócenos, instalaciones, galería, productos, contacto, FAQ y login.

El proyecto fue desarrollado para la empresa “Sexy Lady” que se encuentra ubicada en la calle Aldama #26 en el municipio de Acámbaro, Guanajuato; empresa que se dedica principalmente a la venta de prendas de ropa para mujer, así como bisutería elaborada a mano, entre los principales: aretes, collares y pulseras.

OBJETIVOS GENERALES Y ESPECÍFICOS DEL SISTEMA

Objetivo general

Desarrollar un sitio web con CodeIgniter en PHP durante el periodo Mayo-Agosto 2017 a la empresa **Sexy Lady** para ofrecer a los clientes sus productos; con el fin de ser útil como un medio publicitario para darse a conocer en la región y de esta forma, contar con el control de las ventas del día en un 100%.

Objetivos específicos

1. Planear las actividades para la construcción del sitio web enfocado al manejo de requerimientos y flujos en la administración de proyectos.
2. Recopilar el material necesario para la construcción del sistema web.
3. Analizar la información obtenida con el fin de definir los requerimientos tanto funcionales como no funcionales del sistema.
4. Diseñar los diagramas UML de acuerdo a los requerimientos identificados y especificados durante la fase de análisis del proyecto.
5. Implementar la Base de Datos de acuerdo a los diagramas Entidad-Relación y Relacional obtenidos.
6. Codificar las altas de los diferentes módulos que conformarán al sistema en base a los diagramas UML diseñados.
7. Codificar la función de las altas de los diferentes módulos que conformarán al sistema en base a los diagramas UML diseñados.
8. Codificar la función de las bajas de los diferentes módulos que conformarán al sistema en base a los diagramas UML diseñados.
9. Codificar la función de listar de los diferentes módulos que conformarán al sistema en base a los diagramas UML diseñados.
10. Codificar la función de actualizar de los diferentes módulos que conformarán al sistema en base a los diagramas UML diseñados.

11. Desarrollar las pruebas necesarias al front-end del sitio web para identificar las posibles fallas eventuales que puedan presentarse antes de su implementación.
12. Desarrollar las pruebas necesarias back-end del sitio web para identificar las posibles fallas eventuales que puedan presentarse antes de su implementación.
13. Realizar las pruebas necesarias a los diferentes módulos que conforman el back-end del sitio web para identificar las posibles fallas eventuales que puedan presentarse antes de su implementación.
14. Realizar los ajustes necesarios sobre las posibles fallas presentadas para el buen funcionamiento del sistema.
15. Montar el sitio web a un servidor.
16. Hacer de nuevo las pruebas de integración al sitio web una vez montado al servidor con el fin de que siga en funcionamiento.

NORMAS, POLÍTICAS Y PROCEDIMIENTOS

CMMI

CMMI Acelera la introducción en las organizaciones de producción de software, Mantienen la competitividad entre las organizaciones, Habilitan organizaciones públicas y privadas, para que hagan mejoras en sus prácticas, Fomentan la adopción de estándares de excelencia.

ISO 9001

Está pensada para que, las organizaciones que la apliquen, puedan garantizar su capacidad de ofrecer productos y servicios que cumplen con las exigencias de sus clientes, gracias a una certificación internacional que les brinde prestigio y garantías de calidad.

IEEE 830

El estándar IEEE 830-1998 para el SRS (en inglés) o ERS (Especificación de requerimientos de software) es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.

IEEE 829

Este estándar soporta todos los procesos del ciclo de vida del software, incluyendo la adquisición, suministro, desarrollo, operación y mantenimiento. Este estándar es compatible con todos los modelos de ciclo de vida. No todos los modelos de los ciclos de vida utilizan todos los procesos del ciclo de vida que se enumeran en esta norma.

ISO 27000

Esta norma proporciona una visión general de las normas que componen la serie 27000, indicando para cada una de ellas su alcance de actuación y el propósito de su publicación. Recoge todas las definiciones para la serie de normas 27000 y aporta las bases de por qué es importante la implantación de un SGSI, una introducción a los Sistemas de Gestión de Seguridad de la Información, una breve

descripción de los pasos para el establecimiento, monitorización, mantenimiento y mejora de un SGSI.

DEFINICIÓN DE LAS REGLAS DEL NEGOCIO IMPLEMENTADAS

- La devolución del producto por razones ajenas al negocio, será cubierto por el cliente.
- Para el proceso de devolución se tendrá que hacer una solicitud en la cual se expliquen las razones, para posteriormente la solicitud sea revisada por la administración.
- Los costos del envío están sujetos a la cantidad monetaria comprada.

FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA

En la tecnología web se eligió software del alta calidad como el Lenguaje de programación PHP, De acuerdo a las encuestas de NetCraft, PHP es ahora el módulo más popular para el servidor Apache, creciendo un 4% mensual sobre la totalidad de sitios de Internet. (programacion, 2017)

SGBD MySQL es el gestor de base de datos más popular usado en la red ya que cuenta con distintas ventajas competitivas como libre más rápido y libre, tiene varias herramientas para administración de las bases de datos. (ivancp, 2017)

El servidor web que se utilizo es byethost por que ofrece, 1000 MB (un gigabyte!) De espacio en disco, cuenta FTP y Administrador de archivos, Panel de control, bases de datos MySQL y Soporte PHP, soporte técnico gratuito, dominio adicional, dominios aparcados, subdominios, Comunidad Libre acceso (foros), Los servidores en clúster, ¡Sin anuncios!, https SSL en todos los dominios de alojamiento gratuito. (Certificado autofirmado). (byet.host, 2017)

Se consideró a Codeigneter como el framework a utilizar, porque es muy potente y destaca por su velocidad, sencillez y facilidad de ser aprendido, por lo que resulta ideal para este proyecto, tiene un gran número de librerías y una documentación clara. (miguel, 2017)

Se estableció al IDE NetBeans por que se tomaron en consideración las siguientes características de uso, Netbeans es un entorno de desarrollo gratuito y de código abierto, Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks, optimización de código: por su parte el Profiler nos ayuda a optimizar nuestras aplicaciones e intentar hacer que se ejecuten más rápido y con el mínimo uso de memoria, acceso a base de datos: desde el propio Netbeans podemos conectarnos a distintos sistemas gestores de bases de datos. (genbetadev, 2017)

DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

Actor del negocio	Descripción
Administrador	<p>Este actor estará dedicado a administrar el sitio web.</p> <p>Administrará el contenido de la aplicación web, para mantener actualizados tanto la información como los productos.</p> <p>Algunas características que deberán cumplir los usuarios que administrarán la aplicación son: tener conocimientos sobre el manejo de equipo de cómputo, contar con una previa capacitación en el uso del nuevo sistema, ser ordenado, ser una persona de confianza para la organización, conocer bien el funcionamiento del negocio.</p>
Cliente	<p>Este actor visualiza información además puede consultar y comprar productos publicados en el sitio web.</p> <p>Consultará el contenido de la aplicación y podrá comprar los productos que se ofertan.</p> <p>Algunas características que deberán cumplir los clientes o usuarios que consulten la aplicación son: tener conocimientos sobre el manejo de equipo de cómputo.</p>

ESPECIFICACIÓN DE REQUISITOS

I. Descripción de los Requisitos Funcionales.

ID	Nombre	Descripción
RQF-01	Alta de un producto	<p>Esta función permitirá al usuario capturar los datos necesarios para registrar un nuevo producto en la aplicación web, los cuáles serán los siguientes:</p> <ul style="list-style-type: none">• idProducto• nombre• precio• color• stock• Marca_idMarca• Categoria_idCategoria
RQF-02	Baja de un producto	<p>Esta función permitirá al usuario eliminar uno o más productos que ya no se requieran en la aplicación web, borrando su información como:</p> <ul style="list-style-type: none">• idProducto• nombre• precio• color• stock• Marca_idMarca• Categoria_idCategoria
RQF-03	Editar un producto	<p>Esta función permitirá al usuario modificar los datos necesarios que se requieran hacerlo de uno o más productos que se encuentran en la aplicación, los cuales pueden ser alguno de los siguientes:</p> <ul style="list-style-type: none">• Nombre• precio• color• stock• Marca_idMarca• Categoria_idCategoria
RQF-04	Consultar un producto	<p>Esta función permitirá a la aplicación mostrar al usuario los datos correspondientes de un producto, tales como:</p>

		<ul style="list-style-type: none"> • idProducto • nombre • precio • color • stock • Marca_idMarca • Categoria_idCategoria
RQF-05	Alta de un proveedor	<p>Esta función permitirá al usuario capturar los datos necesarios para registrar un nuevo proveedor en la aplicación web; los cuáles serán los siguientes:</p> <ul style="list-style-type: none"> • idProveedor • nombreProveedor • email • RFC • telefono • Dirección_idDirección
RQF-06	Baja de un proveedor	<p>Esta función permitirá al usuario eliminar uno o más proveedores que ya no se requieran en la aplicación web, borrando su información como:</p> <ul style="list-style-type: none"> • idProveedor • nombreProveedor • email • RFC • telefono • Dirección_idDirección
RQF-07	Editar un proveedor	<p>Esta función permitirá al usuario modificar los datos necesarios que se requieran hacerlo de uno o más proveedores que se encuentran en la aplicación, los cuales pueden ser alguno de los siguientes:</p> <ul style="list-style-type: none"> • nombreProveedor • email • RFC • telefono • Dirección_idDirección
RQF-08	Consultar un proveedor	<p>Esta función permitirá a la aplicación mostrar al usuario los datos correspondientes de un proveedor, tales como:</p> <ul style="list-style-type: none"> • idProveedor • nombreProveedor • email • RFC • Teléfono

		<ul style="list-style-type: none"> • Dirección_idDirección
RQF-09	Alta de un pedido	<p>Esta función permitirá al usuario capturar los datos necesarios para registrar un nuevo pedido en la aplicación web; los cuáles serán los siguientes:</p> <ul style="list-style-type: none"> • idPedido • status • fechaPedido • fechaEntrega • Venta_idVenta • Cliente_idCliente
RQF-10	Baja de una venta	<p>Esta función permitirá al usuario eliminar uno o más pedidos que ya no se requieran en la aplicación web, borrando su información como:</p> <ul style="list-style-type: none"> • idPedido • status • fechaPedido • fechaEntrega • Venta_idVenta • Cliente_idCliente

II. Descripción de los Requisitos No Funcionales.

Listado de Requerimientos del Sistema	
ID	Descripción
RS-001	La aplicación deberá poder ser accesible las 24 horas del día los 365 días del año.
RS-002	El sistema será accesible solo para las personas autorizadas por la empresa.
RS-003	La información capturada por la aplicación deberá ser almacenada de manera segura.
RS-004	El sistema deberá cumplir con los requisitos que desee los(as) responsables de la empresa.
RS-005	El tiempo de respuesta no deberá superar los 15 segundos.
RS-006	La interfaces graficas del sistema deberán tener un diseño adecuado para los usuarios del sistema.
RS-007	El sistema debe ser capaz de operar adecuadamente con hasta 1000 usuarios con sesiones concurrentes
RS-008	El sistema debe respaldarse cada 24 horas.
RS-009	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 3 horas.
RS-010	La aplicación web debe poseer un diseño responsivo con la finalidad de garantizar la adecuada visualización en múltiples computadores

VISTA FUNCIONAL

Tabla 1. Descripción de bajo nivel del caso de uso: CU-01.

CU-01	
Nombre del caso de uso	Alta de un producto.
Actores	<ul style="list-style-type: none">• Administrador.• Sistema.
Resumen	<ol style="list-style-type: none">1. El administrador entra al sistema.2. El sistema muestra la página inicial.3. El administrador introduce su usuario y su contraseña.4. El sistema valida los datos registrados.5. El sistema muestra el panel de administración.6. El administrador ingresa a la sección de productos.7. El sistema muestra la página de productos.8. El administrador selecciona la opción de “Agregar producto”.9. El sistema muestra el formulario para agregar un nuevo producto.10. El administrador introduce los datos necesarios para agregar un nuevo producto.11. El administrador da clic en la opción de “Guardar producto”.12. El sistema valida los datos introducidos.13. El sistema guarda la información introducida en la Base de Datos.14. El sistema muestra la página actualizada de los productos.

Precondiciones	El administrador de la empresa deberá contar con la información necesaria para dar de alta un producto que se desee agregar en el sistema como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo.
Poscondiciones	El administrador tendrá que verificar que el o los productos dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el producto en la lista de productos una vez registrado.
Requisitos especiales	

Tabla 2. Descripción de bajo nivel del caso de uso: CU-02.

CU-02	
Nombre del caso de uso	Baja de un producto.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a eliminar. 9. El administrador da clic en la opción de “Eliminar producto”. 10. El sistema muestra un mensaje de confirmación. 11. El administrador da clic en confirmar para eliminar el producto. 12. El sistema elimina el producto en la Base de Datos. 13. El sistema redirige a la página de productos ya actualizados.
Precondiciones	El administrador tendrá que tener a la mano la información del o los productos que se requieran dar de baja para que estos sean eliminados del sistema como el nombre del producto.
Poscondiciones	El administrador tendrá que verificar que el o los productos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del producto ha sido eliminada.

Tabla 3. Descripción de bajo nivel del caso de uso: CU-03.

CU-03	
Nombre del caso de uso	Editar un producto.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a editar. 9. El administrador da clic en la opción de “Editar producto”. 10. El sistema muestra el formulario para editar un producto con los datos correspondientes a modificar. 11. El administrador modifica los datos necesarios del producto. 15. El administrador da clic en la opción de “Guardar cambios”. 16. El sistema valida los datos modificados. 17. El sistema guarda los datos en la Base de Datos. 18. El sistema redirige a la página de productos ya actualizados.

Precondiciones	El administrador tendrá que tener a la mano la información del o los productos que se requieran editar para que estos puedan ser modificados en el sistema como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo.
Poscondiciones	El administrador tendrá que verificar que el o los productos editados se hayan modificado en el sistema; esto al aparecer la información del producto actualizada en la lista de productos una vez guardada.
Requisitos especiales	

Tabla 4. Descripción de bajo nivel del caso de uso: CU-04.

CU-04	
Nombre del caso de uso	Consultar producto.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a consultar. 9. El administrador da clic en la opción de “Consultar producto”. 10. El sistema muestra una página donde se muestra la información del producto como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo. 11. El administrador consulta a información que necesita del producto. 12. El administrador da clic en la opción “Volver a la lista” para regresar a la lista de productos. 13. El sistema muestra la página donde se encuentran todos los productos listados.

Precondiciones	El administrador tendrá que tener a la mano la información del o los productos que se requieran consultar para que estos puedan ser visualizados en el sistema con su información correspondiente como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo.
Poscondiciones	El administrador tendrá que verificar que el o los productos consultados contengan la información correcta; esto al consultar él seleccionado y ver que el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo sean los verídicos.
Requisitos especiales	

Tabla 5. Descripción de bajo nivel del caso de uso: CU-05.

CU-05	
Nombre del caso de uso	Alta de un proveedor.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de proveedores. 8. El administrador selecciona la opción de “Agregar proveedor”. 9. El sistema muestra el formulario para agregar un nuevo proveedor. 10. El administrador introduce los datos necesarios para agregar un nuevo proveedor. 11. El administrador da clic en la opción de “Guardar proveedor”. 12. El sistema valida los datos introducidos. 13. El sistema guarda la información introducida en la Base de Datos. 14. El sistema muestra la página actualizada de los proveedores.
Precondiciones	El administrador de la empresa deberá contar con la información necesaria para dar de alta un proveedor que se desee agregar en el sistema como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo.

Poscondiciones	El administrador tendrá que verificar que el o los proveedores dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el proveedor en la lista de proveedores una vez registrado.
Requisitos especiales	

Tabla 6. Descripción de bajo nivel del caso de uso: CU-06.

CU-06	
Nombre del caso de uso	Baja de un proveedor.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a eliminar. 9. El administrador da clic en la opción de “Eliminar proveedor”. 10. El sistema muestra un mensaje de confirmación. 11. El administrador da clic en confirmar para eliminar el proveedor. 12. El sistema elimina el proveedor en la Base de Datos. 13. El sistema redirige a la página de proveedores ya actualizados.
Precondiciones	El administrador tendrá que tener a la mano la información del o los proveedores que se requieran dar de baja para que estos sean eliminados del sistema como el nombre del proveedor.

Poscondiciones	El administrador tendrá que verificar que el o los productos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del proveedor ha sido eliminada.
Requisitos especiales	

Tabla 7. Descripción de bajo nivel del caso de uso: CU-07.

CU-07	
Nombre del caso de uso	Editar un proveedor.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a editar. 9. El administrador da clic en la opción de "Editar proveedor". 10. El sistema muestra el formulario para editar un proveedor con los datos correspondientes a modificar. 11. El administrador modifica los datos necesarios del proveedor. 15. El administrador da clic en la opción de "Guardar cambios". 16. El sistema valida los datos modificados. 17. El sistema guarda los datos en la Base de Datos. 18. El sistema redirige a la página de proveedores ya actualizados.

Precondiciones	El administrador tendrá que tener a la mano la información del o los proveedores que se requieran editar para que estos puedan ser modificados en el sistema como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo.
Poscondiciones	El administrador tendrá que verificar que el o los proveedores editados se hayan modificado en el sistema; esto al aparecer la información del proveedor actualizada en la lista de proveedores una vez guardada.
Requisitos especiales	

Tabla 8. Descripción de bajo nivel del caso de uso: CU-08.

CU-08	
Nombre del caso de uso	Consultar proveedor.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<ol style="list-style-type: none"> 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a consultar. 9. El administrador da clic en la opción de “Consultar proveedor”. 10. El sistema muestra una página donde se muestra la información del producto como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo. 11. El administrador consulta a información que necesita del proveedor. 12. El administrador da clic en la opción “Volver a la lista” para regresar a la lista de proveedores. 13. El sistema muestra la página donde se encuentran todos los proveedores listados.

Precondiciones	El administrador tendrá que tener a la mano la información del o los proveedores que se requieran consultar para que estos puedan ser visualizados en el sistema con su información correspondiente como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo.
Poscondiciones	El administrador tendrá que verificar que el o los proveedores consultados contengan la información correcta; esto al consultar él seleccionado y ver que el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo sean los verídicos.
Requisitos especiales	

Tabla 9. Descripción de bajo nivel del caso de uso: CU-09.

CU-09	
Nombre del caso de uso	Alta de un pedido.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<p>19.El administrador entra al sistema.</p> <p>20.El sistema muestra la página inicial.</p> <p>21.El administrador introduce su usuario y su contraseña.</p> <p>22.El sistema valida los datos registrados.</p> <p>23.El sistema muestra el panel de administración.</p> <p>24.El administrador ingresa a la sección de pedidos.</p> <p>25.El sistema muestra la página de pedidos.</p> <p>26.El administrador selecciona la opción de “Agregar pedido”.</p> <p>27. El sistema muestra el formulario para agregar un nuevo pedido.</p> <p>28.El administrador introduce los datos necesarios para agregar un nuevo pedido.</p> <p>29.El administrador da clic en la opción de “Guardar pedido”.</p> <p>30.El sistema valida los datos introducidos.</p> <p>31. El sistema guarda la información introducida en la Base de Datos.</p> <p>32.El sistema muestra la página actualizada de los pedidos.</p>
Precondiciones	El administrador de la empresa deberá contar con la información necesaria para dar de alta un pedido que se desee agregar en el sistema como el id del pedido, el status, la fecha del pedido y la fecha de entre, la venta a la cual pertenece y el cliente a quien se realiza el mismo.

Poscondiciones	El administrador tendrá que verificar que el o los pedidos dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el pedido en la lista de pedidos una vez registrado.
Requisitos especiales	

Tabla 10. Descripción de bajo nivel del caso de uso: CU-10.

CU-10	
Nombre del caso de uso	Baja de un proveedor.
Actores	<ul style="list-style-type: none"> • Administrador. • Sistema.
Resumen	<p>14.El administrador entra al sistema.</p> <p>15.El sistema muestra la página inicial.</p> <p>16.El administrador introduce su usuario y su contraseña.</p> <p>17.El sistema valida los datos registrados.</p> <p>18.El sistema muestra el panel de administración.</p> <p>19.El administrador ingresa a la sección de pedidos.</p> <p>20.El sistema muestra la página de los pedidos listados.</p> <p>21.El administrador localiza el pedido a eliminar.</p> <p>22.El administrador da clic en la opción de “Eliminar pedido”.</p> <p>23.El sistema muestra un mensaje de confirmación.</p> <p>24.El administrador da clic en confirmar para eliminar el pedido.</p> <p>25. El sistema elimina el pedido en la Base de Datos.</p> <p>26.El sistema redirige a la página de pedidos ya actualizados.</p>

Precondiciones	El administrador tendrá que tener a la mano la información del o los pedidos que se requieran dar de baja para que estos sean eliminados del sistema como el id del pedido, el status, la fecha del pedido y la fecha de entre, la venta a la cual pertenece y el cliente a quien se realiza el mismo.
Poscondiciones	El administrador tendrá que verificar que el o los pedidos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del pedido ha sido eliminada.
Requisitos especiales	

VISTA LÓGICA

I. Patrones de diseño.

Un patrón de diseño es un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Existen varias clasificaciones de los patrones, una de ellas es según su finalidad, dentro de la cual se consideran 3 conjuntos fundamentales:

- Patrones de Creación.

Estos patrones crean objetos, evitando dicha función y la instanciación directa por parte de los desarrolladores. Ello proporciona a nuestros programas una mayor flexibilidad para decidir que objetos usar. Dentro de este grupo los más conocidos son: Factory, Abstract Factory, Builder, Prototype y Singleton.

- Patrones Estructurales.

Guían el desarrollo de la aplicación bajo una determinada estructura. Entre ellos se encuentran: Adapter, Bridge, Composite, Decorator, Facade, Flyweight y Proxy.

- Patrones de Comportamiento.

Enfocados principalmente en algoritmos y en la asignación de responsabilidades entre los objetos. Describen el comportamiento de objetos, clases y la comunicación entre ellos.

Estos se clasifican de la siguiente forma:

- **A nivel de clases:** Usan la herencia como vía para distribuir el comportamiento. El Interpreter y el Template Method constituyen ejemplos de este tipo de patrón.
- **A nivel de objetos:** Usan la composición, más la herencia para llevar a cabo las tareas. Muestra de ellos el Command, Iterator, Chain of Responsibility, Strategy y el State.

Patrones de Diseño Detallados

MVC (Modelo Vista Controlador)

a) Descripción del patrón.

Es un patrón de arquitectura de las aplicaciones de software, este patrón separa la lógica de negocio de la interfaz de usuario; así como incrementa la reutilización y flexibilidad.

En pocas palabras, el patrón de diseño MVC organiza el código en base a su función. Por lo que este patrón separa el código en tres capas:

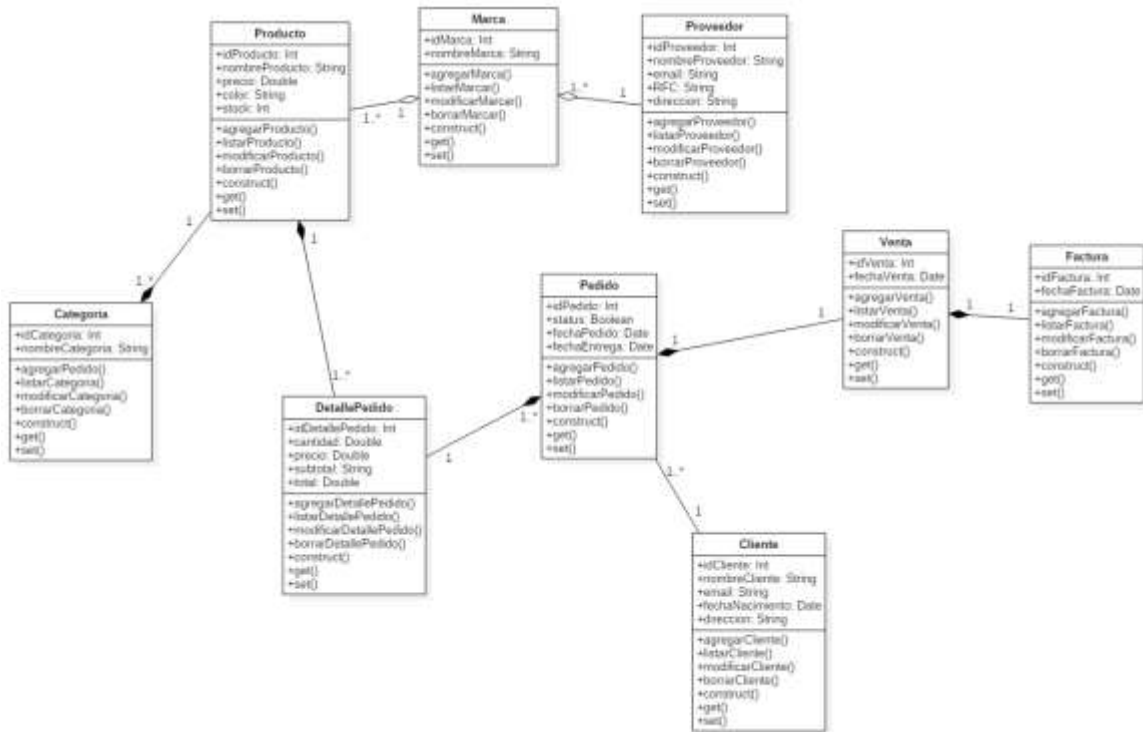
- La capa del **modelo** define la lógica de negocio (la base de datos pertenece a esta capa).
- La **vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa).
- El **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario.

b) ¿Dónde y por qué se emplea?

Este patrón de diseño se emplea en la estructura del proyecto, el cual permite separar la funcionalidad del proyecto de las interfaces del mismo.

Este patrón es empleado en el proyecto, debido a que es un patrón de gran utilidad en el diseño del proyecto integrador, así como en su estructura ya que se contará con un mejor orden de las aceptadas y archivos generados durante el desarrollo del proyecto; permitiendo así la separación de las acciones del usuario y la lógica de la aplicación.

c) Diagrama de clases donde se implementa el patrón.



II. Mecanismos de diseño.

Estos mecanismos son el resultado del refinamiento realizado sobre los mecanismos del análisis. Cabe destacar que este proceso se basa en las restricciones impuestas por el ambiente de implementación.

Algunos ejemplos de estos mecanismos son los siguientes:

- **Mecanismo de seguridad:**

La seguridad es un aspecto importante en el desarrollo de un proyecto ya que se deben contar con las medidas necesarias para evitar posibles vulnerabilidades que puedan presentarse en este mismo.

Así mismo, esta herramienta de seguridad y/o control permite detectar, prevenir o recobrase de alguna vulnerabilidad presentada en el proyecto.

- **Mecanismo de Acceso a Datos:**

El acceso a la información es otro aspecto importante que debe tomarse en cuenta debido a que no todos los usuarios pueden tener acceso a esta misma, por lo que es necesario definir quiénes serán los actores en cada uno de los procesos que son desarrollados en el proyecto.

Mecanismos de Diseño Detallados

a) Descripción de los mecanismos identificados.

- **Mecanismo de seguridad:**

Algunas de las medidas más importantes que se implementarán en el proyecto sobre los mecanismos de seguridad son las siguientes:

- **Restricciones en el sitio:** En el proyecto es importante aplicar restricciones a los usuarios, con la finalidad de dar los permisos necesarios para cada uno de éstos, lo que permitirá que la información sea proporcionada a las personas adecuadas y con sus permisos respectivos sobre la información que se manipula.
- **Rastreo del paso de datos:** Esta medida es importante, ya que debe mantenerse el conocimiento del proceso del manejo de la información; con la finalidad de conocer el origen de los datos, para así mismo identificar si la información que se manipula es de confianza.
- **Filtración de entradas:** Es muy importante el filtrado de la entrada de la información en un proyecto ya que si se realiza esta medida pueden eliminarse riesgos de que los datos dañados sean usados para provocar el mal funcionamiento de la aplicación.

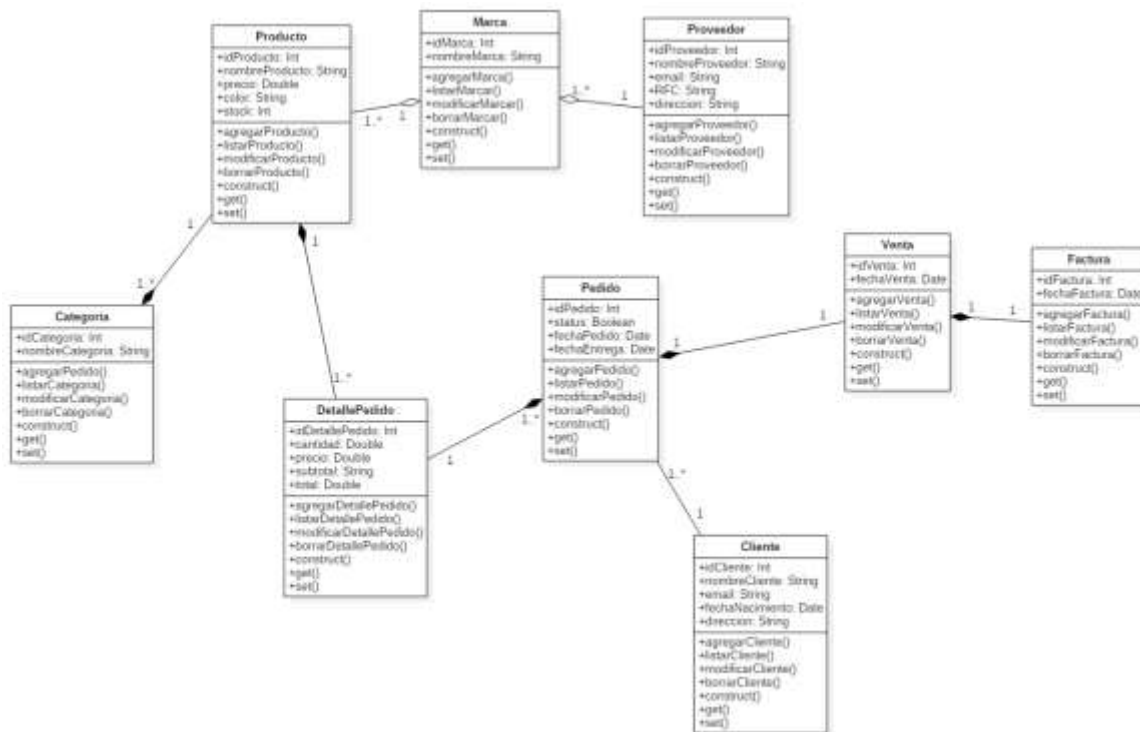
- **Mecanismo de Acceso a Datos:**

Es importante tomar en cuenta la siguiente medida identificada dentro del mecanismo de Acceso a Datos:

- **Diagrama de caso de uso:** Es importante definir cada uno de los procesos que serán ejecutados dentro de la aplicación, debido a que debe tomarse en cuenta los involucrados que participarán en cada proceso y el flujo de actividades que debe seguir para completar este mismo.

b) Documentación de los mecanismos más importantes.

- **Vista estática.**



- **Descripción de cada elemento mostrado en la vista estática.**

El diagrama de clases es la estructura estática que describe la estructura del sistema mostrando sus clases, atributos, operaciones y/o métodos y la relación que hay entre sus objetos; por lo que es importante describir cada una de sus clases:

- ✓ **Categoría:**

Esta clase cuenta con dos atributos: idCategoria (int) y nombreCategoria (String), los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- agregarCategoria(): Método que permitirá al usuario dar de alta una categoría en el sistema.
- listarCategoria(): Método que permitirá al usuario mostrar todas las categorías con las cuales cuenta el sistema.
- modificarCategoria(): Método que permitirá al usuario editar la información de una categoría que se encuentre registrada en la aplicación.
- borrarCategoria(): Método que permitirá al usuario eliminar una categoría que se encuentre registrada en la aplicación.
- construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una categoría registrada en el sistema.

✓ **Producto:**

Esta clase cuenta con dos atributos: idProducto, nombreProducto, precio, color y stock, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- agregarProducto(): Método que permitirá al usuario dar de alta un producto en el sistema.
- listarProducto(): Método que permitirá al usuario mostrar todos los productos con las cuales cuenta el sistema.
- modificarProducto(): Método que permitirá al usuario editar la información de un producto que se encuentre registrado en la aplicación.
- borrarProducto(): Método que permitirá al usuario eliminar un producto que se encuentre registrado en la aplicación.
- construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un producto registrado en el sistema.

✓ **Marca:**

Esta clase cuenta con dos atributos: `idMarca` y `nombreMarca`, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- `agregarMarca()`: Método que permitirá al usuario dar de alta una marca en el sistema.
- `listarMarca()`: Método que permitirá al usuario mostrar todas las marcas con las cuales cuenta el sistema.
- `modificarMarca()`: Método que permitirá al usuario editar la información de una marca que se encuentre registrada en la aplicación.
- `borrarMarca()`: Método que permitirá al usuario eliminar una marca que se encuentre registrada en la aplicación.
- `construct()`, `getters()` y `setters()`: Son los métodos de acceso de cada uno de los atributos con los que cuenta una marca registrada en el sistema.

✓ **Proveedor:**

Esta clase cuenta con dos atributos: `idProveedor`, `nombreProveedor`, `email`, `RFC`, y `dirección`, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- `agregarProveedor()`: Método que permitirá al usuario dar de alta un proveedor en el sistema.
- `listarProveedor()`: Método que permitirá al usuario mostrar todos los proveedores con las cuales cuenta el sistema.
- `modificarProveedor()`: Método que permitirá al usuario editar la información de un proveedor que se encuentre registrado en la aplicación.

- `borrarProveedor()`: Método que permitirá al usuario eliminar un proveedor que se encuentre registrado en la aplicación.
- `construct()`, `getters()` y `setters()`: Son los métodos de acceso de cada uno de los atributos con los que cuenta un proveedor registrada en el sistema.

✓ **DetallePedido:**

Esta clase cuenta con dos atributos: `idDetallePedido`, `cantidad`, `precio`, `subtotal` y `total`, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- `agregarDetallePedido()`: Método que permitirá al usuario dar de alta una detalle de un pedido en el sistema.
- `listarDetallePedido()`: Método que permitirá al usuario mostrar todos los detalles de los pedidos con los cuales se cuentan.
- `modificaDetallePedido()`: Método que permitirá al usuario editar la información de un detalle de pedido que se encuentre registrado en la aplicación.
- `borrarDetallePedido()`: Método que permitirá al usuario eliminar un detalle de un pedido que se encuentre registrado en la aplicación.
- `construct()`, `getters()` y `setters()`: Son los método de acceso de cada uno de los atributos con los que cuenta un producto registrada en el sistema.

✓ **Pedido:**

Esta clase cuenta con dos atributos: idPedido, status, fechaPedido y fecha Entrega, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- agregarPedido(): Método que permitirá al usuario dar de alta un pedido en el sistema.
- listarPedido(): Método que permitirá al usuario mostrar todos los pedidos con las cuales cuenta el sistema.
- modificarPedido(): Método que permitirá al usuario editar la información de un pedido que se encuentre registrado en la aplicación.
- borrarPedido(): Método que permitirá al usuario eliminar un pedido que se encuentre registrado en la aplicación.
- construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un pedido registrado en el sistema.

✓ **Cliente:**

Esta clase cuenta con dos atributos: idCliente, nombreCliente, email, fechaNacimiento y direccion, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- agregarCliente(): Método que permitirá al usuario dar de alta un cliente en el sistema.
- listarCliente(): Método que permitirá al usuario mostrar todos los clientes con las cuales cuenta el sistema.
- modificarCliente(): Método que permitirá al usuario editar la información de un cliente que se encuentre registrado en la aplicación.
- borrarCliente(): Método que permitirá al usuario eliminar un cliente que se encuentre registrado en la aplicación.

- `construct()`, `getters()` y `setters()`: Son los métodos de acceso de cada uno de los atributos con los que cuenta un cliente registrada en el sistema.

✓ **Venta:**

Esta clase cuenta con dos atributos: `idVenta` y `fechaVenta`, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- `agregarVenta()`: Método que permitirá al usuario dar de alta una venta en el sistema.
- `listarVenta()`: Método que permitirá al usuario mostrar todas las ventas con las cuales cuenta el sistema.
- `modificarVenta()`: Método que permitirá al usuario editar la información de una venta que se encuentre registrada en la aplicación.
- `borrarVenta()`: Método que permitirá al usuario eliminar una venta que se encuentre registrada en la aplicación.
- `construct()`, `getters()` y `setters()`: Son los métodos de acceso de cada uno de los atributos con los que cuenta una venta registrada en el sistema.

✓ **Factura:**

Esta clase cuenta con dos atributos: idFactura y FechaFactura, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

- agregarFactura(): Método que permitirá al usuario dar de alta una factura en el sistema.
- listarFactura(): Método que permitirá al usuario mostrar todas las facturas con las cuales cuenta el sistema.
- modificarFactura(): Método que permitirá al usuario editar la información de una factura que se encuentre registrada en la aplicación.
- borrarFactura (): Método que permitirá al usuario eliminar una factura que se encuentre registrada en la aplicación.
- construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una factura registrada en el sistema.

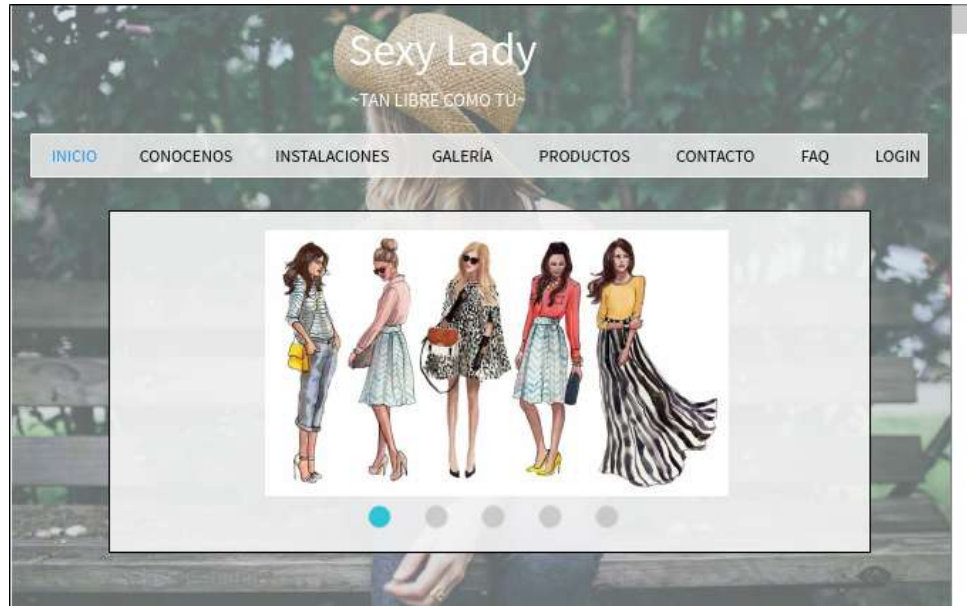
MODELO LÓGICO DE DATOS

La relación existente entre el diagrama de clases y el diagrama relacional es que cuentan con características similares; las cuales se describen a continuación:

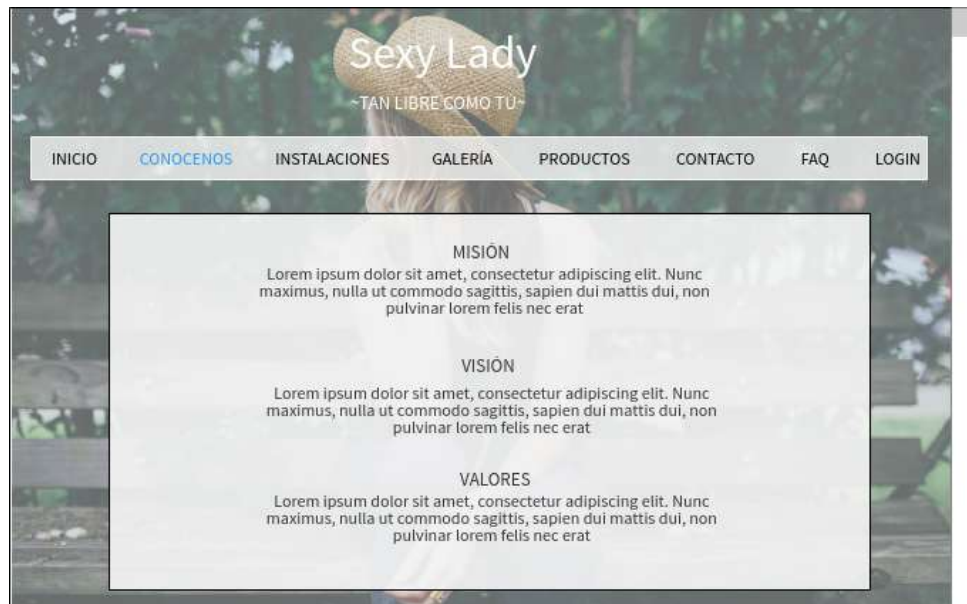
- **Entidades:** En base al diagrama E-R se diseñó el diagrama relacional, el cual cuenta con los mismos atributos; pero una vez normalizado se agregaron nuevas tablas al modelo relacional con la finalidad de que la información no sea tan redundante. Mientras que en el diagrama de clases cuenta con los mismos atributos definidos en el diagrama E-R.
- **Llaves primarias:** En cada una de las tablas que conforman los diagramas, tanto el diagrama relacional como el diagrama de clases cuentan con una llave primaria, la cual será el identificador de cada uno de los registros que se realicen en estas mismas.
- **Llaves foráneas:** De igual forma, las llaves foráneas se representan en el diagrama relacional donde se pueden apreciar de forma visible al unir dos tablas como se representan de acuerdo a la cardinalidad definida en el diagrama E-R, y al ser una relación de muchos a muchos se puede visualizar una tercera tabla donde se muestran las llaves foráneas de ambas tablas; de lo contrario, en el diagrama de clases no se encuentran representadas aunque si se puede visualizar la asociación que hay entre cada una de las tablas.

DISEÑO DE PANTALLAS

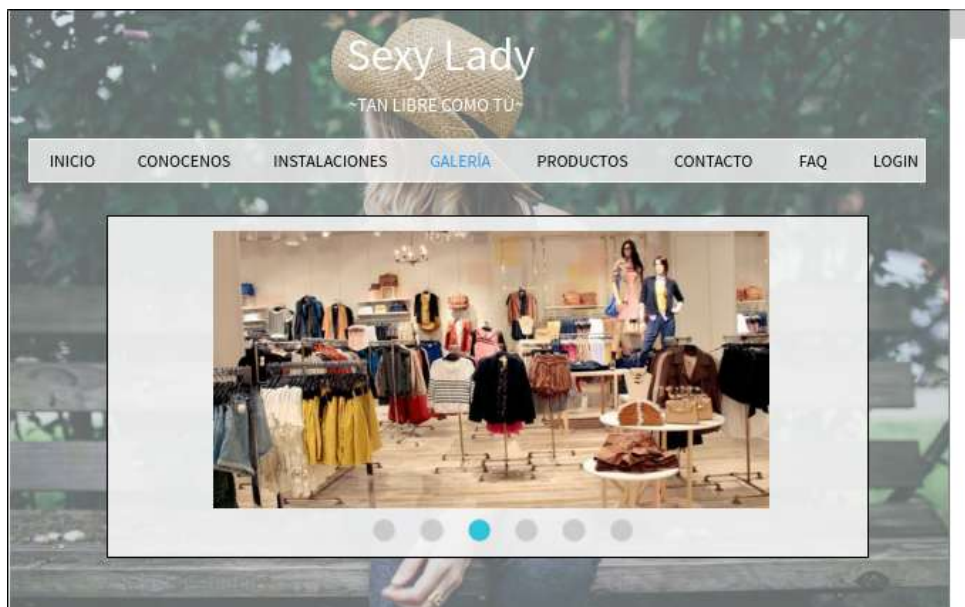
- Página de “Inicio”.



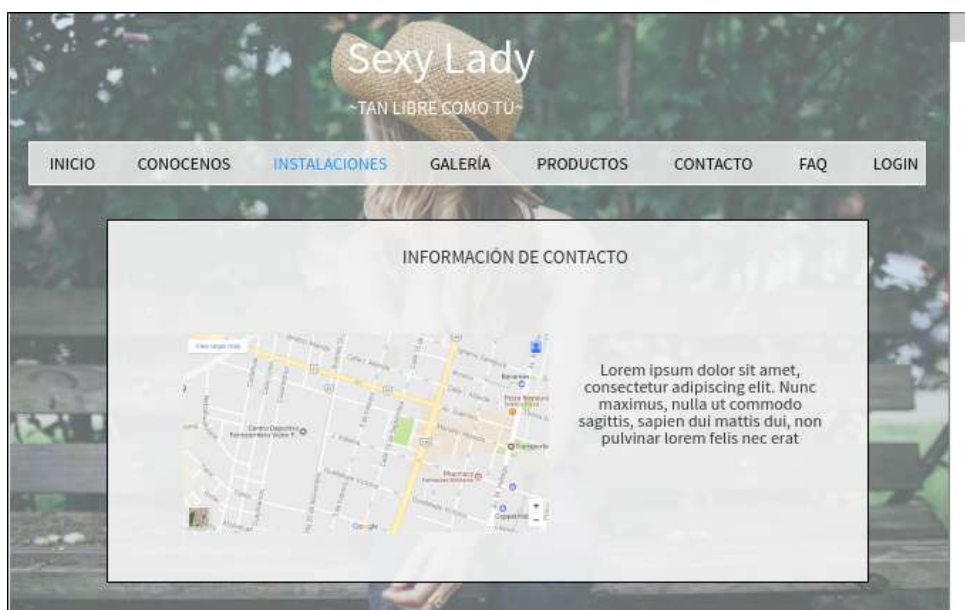
- Página de “Conócenos”.



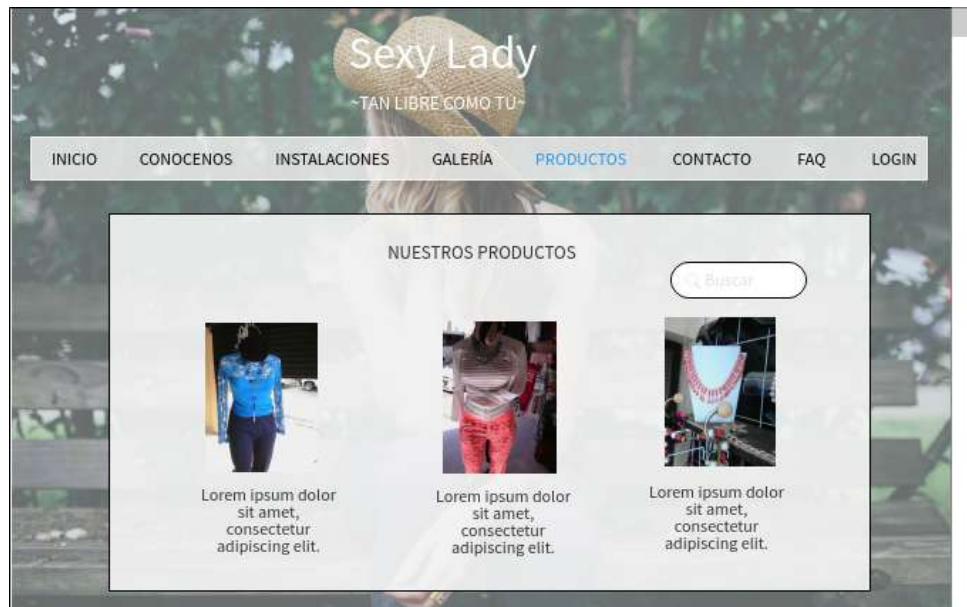
- Página de “Galería”.



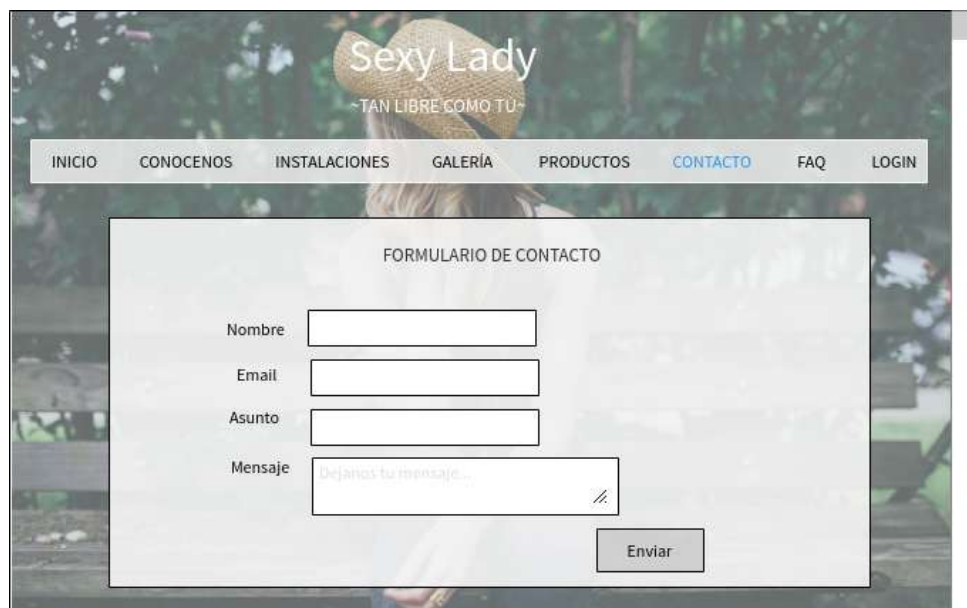
- Página de “Instalaciones”.



- Página de “Productos”.



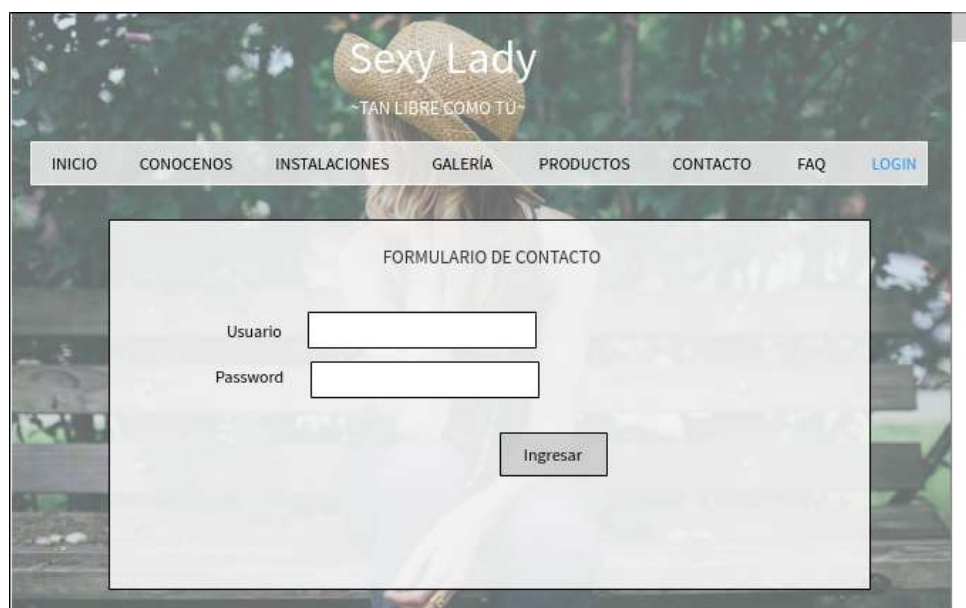
- Página de “Contacto”.



- Página de “FAQ”.



- Página de “Login”.

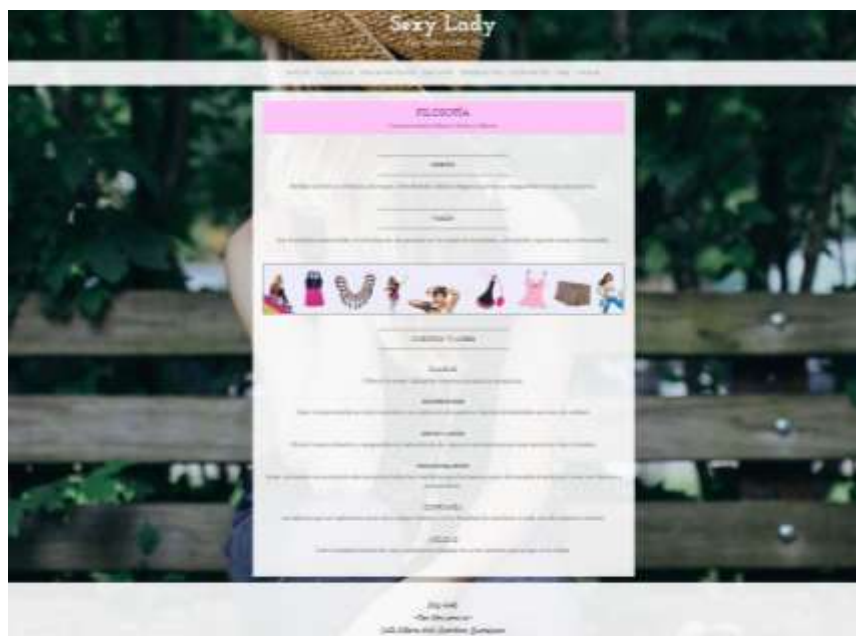


DESCRIPCIÓN DE CAMPOS REQUERIDOS POR PANTALLA

Pantalla Inicio



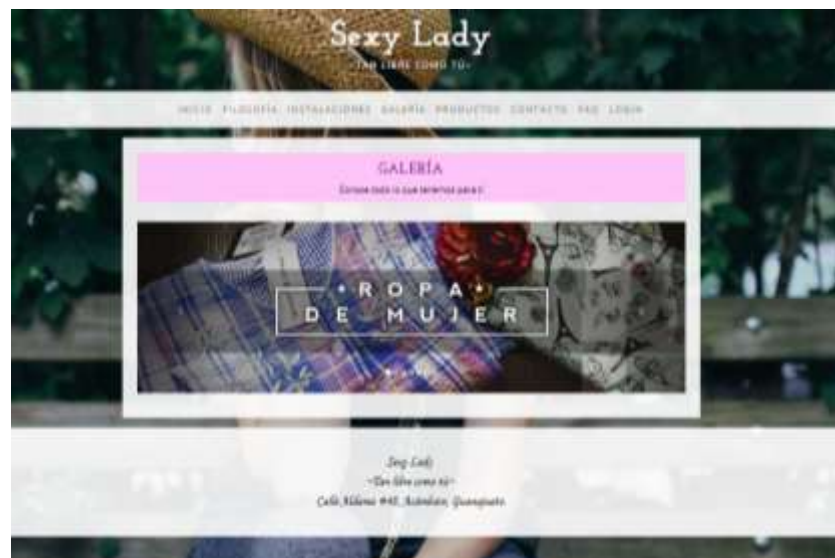
Pantalla Filosofía



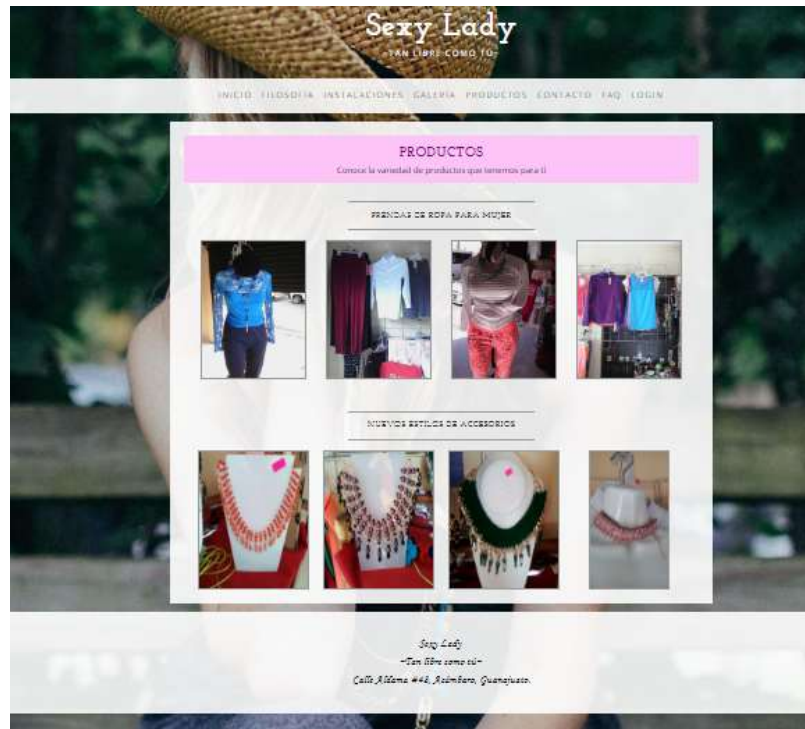
Pantalla Instalaciones



Pantalla Galería



Pantalla Productos



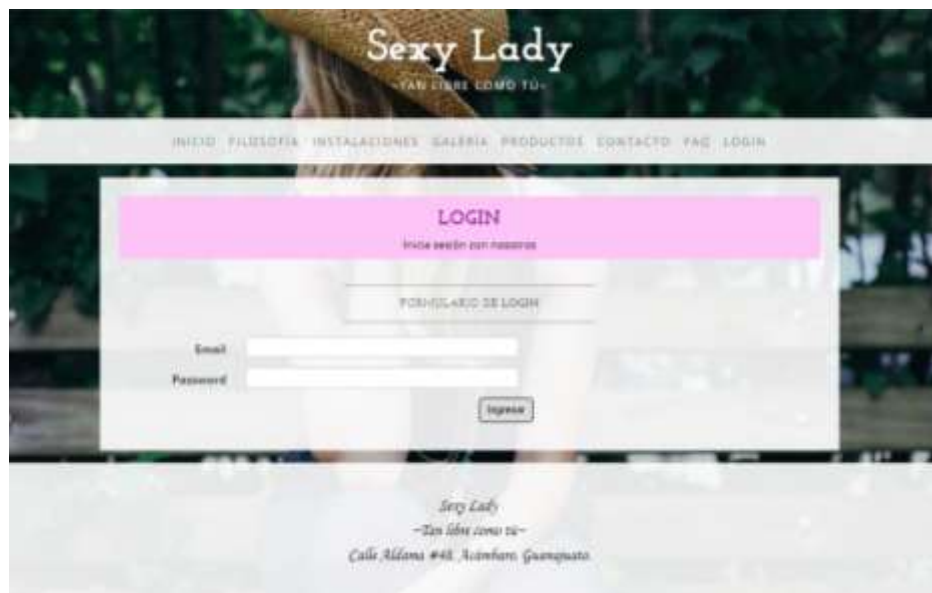
Pantalla Contáctanos

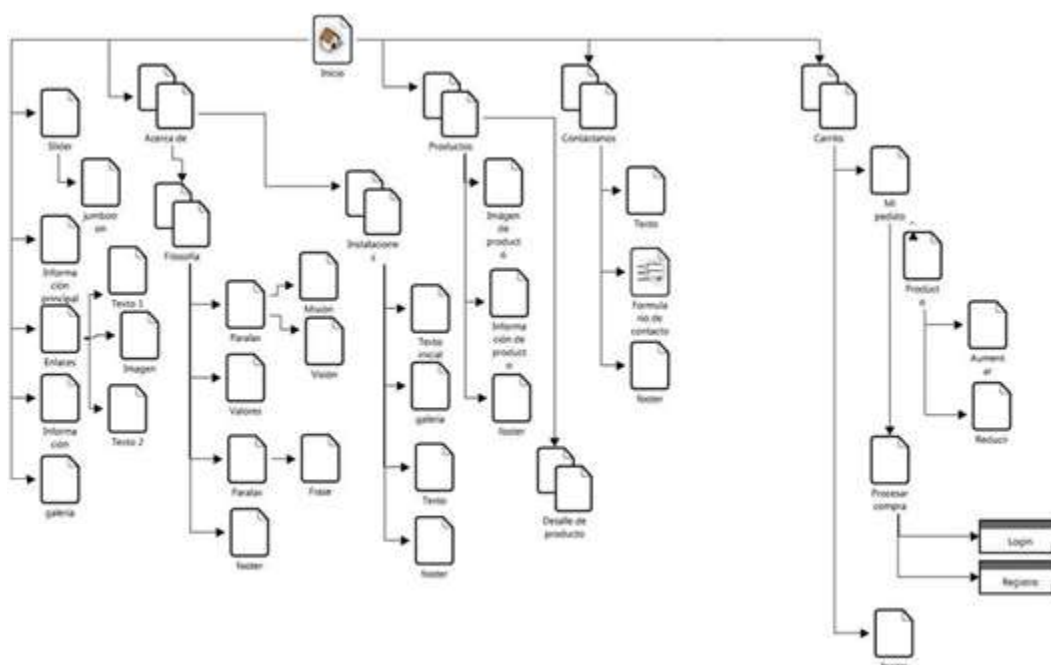


Pantalla FAQ



Pantalla Login





VISTA DE DESPLIEGUE

El diagrama de despliegue muestra la configuración en funcionamiento del sistema incluyendo el software y hardware, además modela la arquitectura en tiempo de ejecución del sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos del software se trazan en esos nodos.

Un Nodo es un elemento de hardware o software como se muestran en el diagrama y una asociación representa una ruta de comunicación entre estos nodos. El siguiente diagrama muestra un diagrama de despliegue para un sitio web, mostrando los nodos necesarios y la asociación.

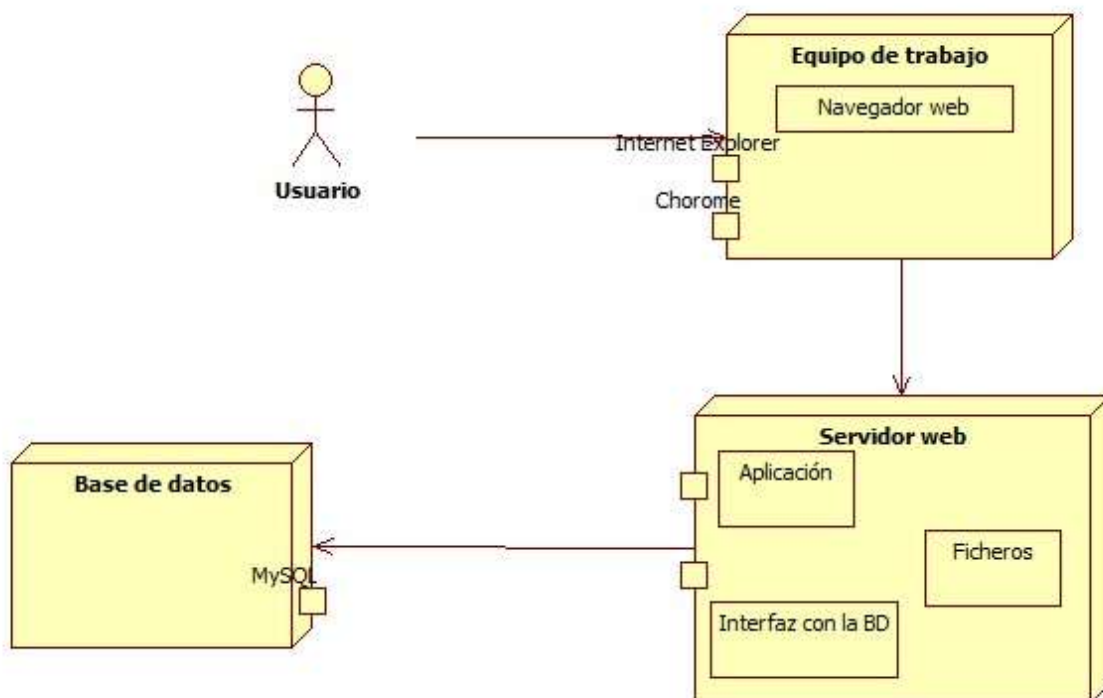
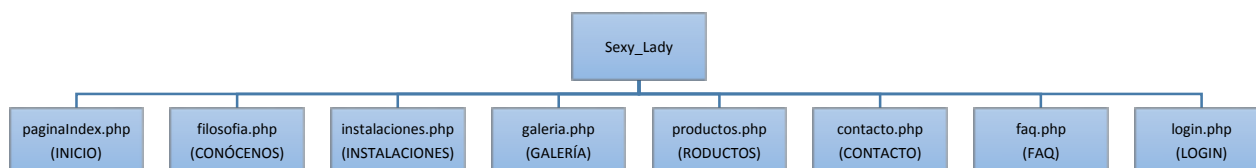


DIAGRAMA DE NAVEGACIÓN DEL SISTEMA



GLOSARIO DE TÉRMINOS

Administrador de proyecto

El administrador de proyecto es la persona que administra y controla los recursos asignados a un proyecto, con el propósito de que se cumplan correctamente los planes definidos (Fuller Padilla, 2016).

Análisis

En esta etapa se definen los grandes rasgos del sistema de software que tendrá que dar soporte informático a unas actividades determinadas de unos ciertos usuarios dentro del marco más general de la actividad de la empresa u organización (Campderrich Falgueras, 2016).

Analista

La palabra “análisis” se refiere a una característica típicamente relacionada con la inteligencia humana. Esta se refiere a la habilidad de poder estudiar un problema de una complejidad determinada, descomponiendo el problema en sub-problemas de menor complejidad. De esa forma, la solución del problema completo se obtiene como la suma de las soluciones de los sub-problemas de menor complejidad (Fuller Padilla, 2016).

Base de Datos

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

La base de datos es un conjunto de datos organizados en estructuras que se definen una sola vez y que se utilizan al mismo tiempo por muchos equipos o usuarios (Hueso Ibañez, 2016).

CASE

CASE significa Computer-Aided Software Engineering. Las herramientas CASE son software de apoyo al desarrollo, mantenimiento y documentación informatizados de software (Campderrich Falgueras, 2016).

Caso de secuencia

Con los diagramas de secuencia se representa la interacción de los objetos (envío de mensajes), que permite visualizar cuál es la secuencia de operaciones (métodos) a realizar para resolver el conjunto de pasos indicados en el flujo normal y alternativo de la documentación de los casos de uso (Coronel, 2010).

Caso de uso

Es uno de los principales diagramas de UML que permite representar, analizar y documentar los requerimientos funcionales del software. Y está compuesto por los siguientes elementos Actores, Casos de uso y Relaciones (Comunicación, Extensión, Inclusión y Generalización)” (Coronel, 2010).

Ciclo de vida

Es una descripción simplificada de un proceso del software que presenta una visión de este proceso (Sommerville, 2005). Este modelo puede incluir actividades que son partes de los procesos y productos de software y el papel de las personas involucradas.

Diagrama de actividades

Este diagrama se utiliza para representar el flujo de actividades de un negocio, caso de uso o una operación, método (Coronel, 2010).

El diagrama de actividades se puede considerar una variante tanto del diagrama de estados como de los diagramas de interacción, ya que sirve para describir los estados de una actividad.

Diagrama de clases

El Ing. Eric Gustavo Coronel Castillo define que “Los diagramas de clases representan la implementación que tendrá el software y que pueden crearse diagramas de clases enfocado al modelo de datos y al modelo de sistemas” (Coronel, 2010).

Diagrama Entidad/Relación

Los diseños de bases de datos suelen apoyarse en diagramas a través de los cuales se tratan de visualizar las diferentes entidades que intervienen, las relaciones entre ellas y el tipo de estas relaciones. Estos gráficos son de ayuda para decidir las distintas tablas que deben ser utilizadas en la base de datos (Cobo, 2016).

Diseñador

Es el encargado de generar el diseño del sistema (Fuller Padilla, 2016).

Diseño

La etapa de diseño es el mejor momento para elaborar la Especificación de la prueba, que describe con qué datos se tiene que probar cada programa o grupo de programas y cuáles son los resultados esperados en cada caso (Campderrich Falgueras, 2016).

Modelo relacional

Este modelo se basa en representar los datos mediante tablas con diferentes atributos a modo de columnas. La existencia de atributos comunes en las tablas permite establecer relaciones entre ellas (Cobo, 2016).

MDL

El MDL (Manipulation Data Language) es la parte de SQL dedicada a la manipulación de los datos, es decir, inserción, borrado, modificación y consulta de los mismos (Hueso Ibañez, 2016).

MySQL

MySQL es un sistema de administración de bases de datos relacionales rápido, sólido y flexible. Es ideal para crear bases de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones on-line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas (Cobo, 2016).

Programación

La programación o codificación, es la etapa que consiste en traducir el diseño a código procesable por el ordenador. Es en esta etapa donde se le da forma real al software, es en realidad cuando se elabora. El entregable que se genera en esta etapa es el programa propiamente, con todas sus funcionalidades y componentes (Campderrich Falgueras, 2016).

Programador

Los programadores deben convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación (Fuller Padilla, 2016).

Prueba

La etapa de prueba consiste en probar el software desde distintos puntos de vista de una manera planificada y, naturalmente, localizar y corregir dentro del software y su documentación los errores que se detecten (Campderrich Falgueras, 2016).

Requerimiento

Es una descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, realizar un pedido o consultar información (Sommerville, 2005).

SQL

SQL es un lenguaje de definición y manipulación de datos para bases de datos relacionales. Es un lenguaje de definición porque permite definir la estructura de las tablas que componen la base de datos, y de manipulación porque permite efectuar consultas y realizar operaciones como inserción, borrado y actualización de los datos que contiene (Cobo, 2016).

TÉSTER

El téster es el encargado de asegurar la calidad de cada uno de los productos (documentos, prototipos, etc) (Fuller Padilla, 2016).

UML

El Unified Modeling Language (UML) es un modelo para la construcción de software orientado a objetos que ha sido propuesto como estándar de ISO por el OMG. Consta de un conjunto de tipos de diagramas interrelacionados, dentro de los cuales se utilizan elementos del modelo, que sirven para describir distintos aspectos de la estructura y la dinámica del software (Campderrich Falgueras, 2016).

ESTÁNDARES DE ELABORACIÓN DEL MANUAL

Toda la documentación que se relacione con un sistema, ya sea impresa o digital, sencilla o compleja, debe reunir los siguientes requisitos básicos:

- Debe ser rotulada con claridad y bien organizada en carpetas e índice, con secciones claramente indicadas.
- Los diagramas deberán ser claros, no aglomerados y la escritura manuscrita ha de ser legible.
- La documentación deberá ser completa.
- Se incluirá una leyenda o explicación de los términos utilizados.
- La documentación siempre se conserva actualizada.

El estilo de redacción de los manuales de documentación debe:

- Ser concreto.
- Definir los términos utilizados.
- Utilizar títulos, subtítulos y párrafos cortos.
- Emplear formas activas en lugar de pasivas.
- Aplicar correctamente las referencias bibliográficas.
- No usar frases largas que presenten hechos distintos.

3.1 Estándares que son aplicados en el proyecto

Las normas y/o estándares que serán aplicados al proyecto integrador son los siguientes:

- **CMMI.** Esta norma y/o estándar puede ser aplicada en el proyecto, ya que se cuenta con una línea base de proyectos anteriores, por lo cual se cuenta con la mayoría de las plantillas que corresponden al llenado de la documentación del proyecto y que éstas a su vez ya han sido auditadas para llevar a cabo un proceso de calidad durante el desarrollo de cada uno de los proyectos que se realice. Además, este modelo evalúa cada uno de los procesos que son realizados dentro de la organización.
- **ISO 9001.** Esta norma y/o estándar nos permitirá llevar a cabo un buen sistema de calidad durante las diferentes actividades que se desarrollarán durante todo el proceso del proyecto y de esta forma, se podrá mantener la información del proyecto de forma completa y en orden cumpliendo con cada uno de los requisitos que se requieren para contar con la documentación estandarizada y de esta manera mejorar la calidad del proyecto de forma eficaz y eficiente.
- **IEEE 830.** Esta norma y/o estándar nos permitirá conocer de forma detallada cada uno de los requerimientos que nos proporcionará el cliente sobre el proyecto a través del **SRS** (Software Requirements Specification) o ERS (Especificación de Requerimientos de Software); así mismo, a aplicar esta norma ayuda al cliente a describir de forma clara los requerimientos que debe cumplir el software por lo que es muy importante contar con la participación el cliente durante el desarrollo del proyecto ya que durante el proceso del mismo pueden surgir cambios por parte del cliente que no se adecuen a sus necesidades por lo que éste a su vez va determinando claramente lo que requiere o no, en el software.

- **IEEE 829.** Esta norma y/o estándar puede ser aplicada en el proyecto ya que el ciclo de vida seleccionado para el desarrollo del mismo incluye las etapas que mejor se adecuan a este estándar; además proporciona un conjunto estandarizado de documentos para la documentación del software lo que permite preparar las pruebas con los documentos correspondientes y de esta forma, ejecutarlas para así mismo obtener el reporte de todas aquellas que son realizadas para contar con un sistema en buen funcionamiento y de calidad.
- **ISO 27000.** Esta norma y/o estándar nos proporcionará una visión general de cada una de las normas que conforman a esta misma, así como de los sistemas de gestión de la seguridad de la información; lo que permitirá administrar y/o gestionar toda la información de la empresa a quién se desarrolla el proyecto y de esa forma, mantener protegida la información más crítica y vulnerable de la compañía.